

논 단 / 자 료

SOLARS 유니코드 기반 시스템 구축을 위한 기초 연구

김 미 향
정보관리과 정보화지원실

< 목 차 >

- | | |
|----------------------|-----------------------|
| 1. 서언 | 4.2. 외부기관과 데이터 교류 방안 |
| 1.1. 다국어 문자 표현 방법 | 4.3. 다국어 입력 방식 |
| 1.2. 유니코드 | 4.4. 유니코드 인코딩 방식 |
| 2. 국내외 유니코드 체계 적용 사례 | 4.5. 유니코드 미지원 한자 처리 |
| 2.1. 국외 | 4.6. 한글음이 없는 한자 입력 방법 |
| 2.2. 국내 | 4.7. 옛한글 문제 |
| 3. 유니코드 적용의 필요성 | 4.8. 검색엔진 교체 |
| 4. 유니코드 변환시 고려사항 | 5. 서울대 도서관 적용 방안 |
| 4.1. 동형이음 한자 처리 | 6. 결론 |

1. 서언

전세계에는 다양한 국가와 민족이 살고 있고, 수많은 언어와 문자가 존재한다. 이러한 언어와 문자를 컴퓨터상에서 표현할 때 서로 다른 문자세트(Character Set)와 인코딩(Encoding)방식을 사용하게 된다. 이로 인해 국가간, 기관간 자료의 상호 교환이나 다국어를 입력할 때 문제가 야기될 수 있으며 또한 입력할 수 있는 문자수의 한계로 인하여 문자표현에 어려움을 겪을 수밖에 없다. 우리의 경우 KSC 5601이라는 문자세트를 사용하고 있으나 중국은 GB코드, 일본의 경우에는 JIS코드 문자세트를 채택하고 있다. 따라서 각국에서 작성된 문서나 자료를 서로 교환하고자 할 때 글자가 깨져서 보이는 등의 문제가 발생하게 된다. 이러한 문제로 인해 여러

개의 소프트웨어 업체들이 모여 ‘지구상의 모든 문자를 표현할 수 있는 문자세트를 만들어 보자’라고 해서 만들어지게 된 것이 바로 유니코드(Unicode)이다.

1.1 다국어 문자 표현 방법

초기 컴퓨터 시스템은 기본적으로 미국표준협회(ANSI, American National Standard Institute)에서 제정한 ASCII 코드에 기초하여, 영어 알파벳 26자와 약간의 특수기호만을 표현할 수 있었다. 이 시스템은 연산 속도 및 기억용량의 한계를 가지고 있었고, 동양권 언어를 전혀 고려하지 않아 이 정도의 성능과 정보 표현방식으로도 충분한 성능을 발휘하였다. 그러나 1바이트로 구성된 ASCII 코드로 모든 문자를 표현할 수 없는 언어를 사용하는 국가에서는 각각 자신들만의 인코딩 방법을 이용하여 문자를 표현하게 되었다.

1.1.1 ASCII 코드

ASCII는 American Standard Code for Information Interchange의 약자로 근대 영어와 다른 서구 유럽 언어에서 사용되는 로마 알파벳에 기초한 문자세트와 문자 인코딩 방식으로, 텍스트를 표현하기 위한 컴퓨터, 통신 장비, 제어장치에서 가장 많이 사용되고 있다.

1950년 후반 미국표준화연합(ANSI의 전신)이 데이터의 저장과 교환에 사용하기 위하여 새로운 코드를 개발하였다. 이 당시의 컴퓨터는 유닉스 기반의 컴퓨터로 7비트가 1바이트인 메모리를 사용하였다. 따라서 자연스럽게 하나의 문자를 표현하는데 7비트를 사용함에 따라 128개의 문자를 표현할 수 있었다. 널(null)로 시작하는 32개 제어문자, 공백문자(space), 94개 프린트문자, 지움문자(delete)로 128개의 비트 조합을 배정하였다. 이에 따라 1963년 최초의 ASCII 표준이 제정되었다. 그런데 이때까지만 해도 많은 비트 조합이 문자에 배정되지 않은 상태였으며, 1968년에 오늘날과 같은 ASCII 코드가 만들어져 표준으로 지정되었다.

1.1.2 ISO 8859

다국어 표현에 적합한 문자세트를 만들려는 국제적인 노력은 ISO 8859시리즈에서 나타났다. 이런 9가지 문자세트(8859-1 ~ 8859-10에서 8859-4는 개정을 위해 철회됨)는 서유럽, 동유럽, 아라비아, 그리스 등의 언어를 표현할 수 있게 했다. ISO

8859-1은 서유럽 언어를 다루기에 충분했으며, ISO 8859-7은 기본적인 그리스 문자, 알파에서 오메가, 발음기호 등 현대의 그리스 문자와 영어를 모두 지원했다.

ISO 8859에서는 하나의 문자를 표현하는데 8비트를 사용하며 ASCII 외의 각 나라마다 자신이 사용할 문자는 128 이후에 배치하였다.

1.1.3 ISO 2022

ISO 2022는, 공식적으로는 ISO/IEC 2022로서, 2바이트 인코딩을 이용하지 않고 단일문자 인코딩(a single character encoding)으로 복합 문자 정보(multiple character sets)를 표현할 수 있도록 개발된 기술이 ISO 2022이다. 각 문자당 8비트를 사용한 ISO 8859 문자 인코딩과 달리, ISO 2022 인코딩은 각 문자당 8비트 혹은 16비트를 사용하는 가변길이 인코딩(variable size encoding)을 전형적으로 사용한다.

그리스어, 러시아어, 아랍어 또는 히브리어 등과 같이 라틴 알파벳에 기반을 두지 않은, 많은 언어와 어족들은 역사적으로 문자세트의 ISO 8859를 포함하는 8비트 확장 ASCII 인코딩으로 컴퓨터에서 표현되어 왔다. 중국어, 일본어, 한국어와 같은 동아시아 문자들은 8비트 컴퓨터에서 표현 가능한 문자보다 더 많은 문자들을 이용하므로, 처음에는 언어 종속적인 2바이트 인코딩(language-specific double byte encoding)으로 컴퓨터에서 표현되었다. ISO 2022 문자 인코딩들은 이어서 나오는 문자들에 대한 문자세트를 지시하는 이스케이프 시퀀스(escape sequence)를 포함하고 있다.

ISO/IEC 2022에서는 코드 테이블의 특정 위치에 특정문자, 01/11에 ESCAPE(ESC), 02/00에 SPACE(SP), 07/15에 DELETE(DEL)를 배정해 두었다. ESC 문자의 위치는 ASCII가 만들어질 때부터 있었으나 위치가 고정될 정도로 중요성이 부각된 일은 최근 일이다. Coded character sets(코드화된 문자셋의 확장)이 발전함에 따라 '변수를 포함하는 제어 정보'를 코드화할 필요성이 부각되었고, 이를 위해 처음에 ESC문자를 사용한다. 이와 같이 제어정보를 코드화하여 나타내는 것을 이스케이프 시퀀스라고 한다.

ISO 2022 문자세트들은 여전히 많이 사용되고 있으나, 최근에는 최신 이메일 소프트웨어 등에서 UTF-8과 같은 유니코드 문자 인코딩을 사용하는 것으로 차츰 변환되고 있다.

1.1.4 KSC 5601

KSC 5601은 94x94의 각 위치(행열)에 한글 문자를 일정한 순서에 따라 배열해 놓은 문자세트를 의미한다. 한글 코드의 KS 제정에서 완성형이 채택된 것은 내부적으로 한글의 출력이 모아쓰기 형태로 이루어지면서 한자를 섞어서 쓸 수 있어야 한다는 사회적 요구로 조합형을 수용하기가 어려웠기 때문이다. 또 다른 배경은 국가 간의 정보교환을 위한 코드 표준화 과정에서 ISO 2022에서 제정한 코드 체계에 따라 세계 각국의 문자를 처리하는데 기인한다. 이는 1바이트 코드로 한 문자 표현이 불가능한 CJK(Chinese, Japanese, Korean) 문자를 2바이트 코드 영역의 첫 번째 영역에 넣을 수 있도록 영역을 확보해야 했기 때문이다.

이와 같은 배경에 의해 KSC 5601-1982에 의한 2바이트 조합형 코드가, 1987년에 2바이트 완성형 코드인 정보교환용 부호에 관한 한글 공업 규격으로 새로 바뀌게 된 것이다.

KSC 5601은 완성형 한글 2,350자, 한자 4,888자, 기술/학술기호 등 특수문자 432자, 숫자 30자, 한글 낱자 94자, 로마문자 52자, 그리스 문자 48자, 괄선 조각 68자, 라틴 문자 27자, 일본 문자 169자, 러시아 문자 66자 등 총 8,224자와 기타 사용자 정의 영역으로 한글 96자, 한자 95자 정도를 사용하도록 배정하고 있다.

1.2 유니코드

1.2.1 개요

유니코드(Unicode)는, 컴퓨터에 저장하려는 모든 문서의 텍스트에 대해 인코딩 방법을 제공하기 위한 국제 표준이다. 유니코드는 오늘날 실제 사용되는 모든 문자와, 학자만 사용하는 문자, 수학기호, 언어학기호, 프로그래밍 언어 기호 등까지 포함하고 있다.

[표 1] 유니코드 3.1에 정의된 언어 영역 (List of block names for Unicode Standard 3.1)

범위	언어	범위	언어
0000..007F	Basic Latin	2460..24FF	Enclosed Alphanumerics
0080..00FF	Latin-1 Supplement	2500..257F	Box Drawing
0100..017F	Latin Extended-A	2580..259F	Block Elements
0180..024F	Latin Extended-B	25A0..25FF	Geometric Shapes
0250..02AF	IPA Extensions	2600..26FF	Miscellaneous Symbols

02B0..02FF	Spacing Modifier Letters	2700..27BF	Dingbats
0300..036F	Combining Diacritical Marks	2800..28FF	Braille Patterns
0370..03FF	Greek	2E80..2EFF	CJK Radicals Supplement
0400..04FF	Cyrillic	2F00..2FDF	Kangxi Radicals
0530..058F	Armenian	2FF0..2FFF	Ideographic Description Char.
0590..05FF	Hebrew	3000..303F	CJK Symbols and Punctuation
0600..06FF	Arabic	3040..309F	Hiragana
0700..074F	Syriac	30A0..30FF	Katakana
0780..07BF	Thaana	3100..312F	Bopomofo
0900..097F	Devanagari	3130..318F	Hangul Compatibility Jamo
0980..09FF	Bengali	3190..319F	Kanbun
0A00..0A7F	Gurmukhi	31A0..31BF	Bopomofo Extended
0A80..0AFF	Gujarati	3200..32FF	Enclosed CJK Letters and Months
0B00..0B7F	Oriya	3300..33FF	CJK Compatibility
0B80..0BFF	Tamil	3400..4DB5	CJK Unified Ideographs Ext. A
0C00..0C7F	Telugu	4E00..9FFF	CJK Unified Ideographs
0C80..0CFF	Kannada	A000..A48F	Yi Syllables
0D00..0D7F	Malayalam	A490..A4CF	Yi Radicals
0D80..0DFF	Sinhala	AC00..D7A3	Hangul Syllables
0E00..0E7F	Thai	D800..DB7F	High Surrogates
0E80..0EFF	Lao	DB80..DBFF	High Private Use Surrogates
0F00..0FFF	Tibetan	DC00..DFFF	Low Surrogates
1000..109F	Myanmar	E000..F8FF	Private Use
10A0..10FF	Georgian	F900..FAFF	CJK Compatibility Ideographs
1100..11FF	Hangul Jamo	FB00..FB4F	Alphabetic Presentation Forms
1200..137F	Ethiopic	FB50..FDFF	Arabic Presentation Forms-A
13A0..13FF	Cherokee	FE20..FE2F	Combining Half Marks
1400..167F	Unified Canadian Aboriginal Sylla.	FE30..FE4F	CJK Compatibility Forms
1680..169F	Ogham	FE50..FE6F	Small Form Variants
16A0..16FF	Runic	FE70..FEFE	Arabic Presentation Forms-B
1780..17FF	Khmer	FEFF..FEFF	Specials
1800..18AF	Mongolian	FF00..FFEF	Halfwidth and Fullwidth Forms
1E00..1EFF	Latin Extended Additional	FFF0..FFFD	Specials
1F00..1FFF	Greek Extended	10300..1032F	Old Italic
2000..206F	General Punctuation	10330..1034F	Gothic
2070..209F	Superscripts and Subscripts	10400..1044F	Deseret
20A0..20CF	Currency Symbols	1D000..1D0FF	Byzantine Musical Symbols
20D0..20FF	Combining Marks for Symbols	1D100..1D1FF	Musical Symbols
2100..214F	Letterlike Symbols	1D400..1D7FF	Mathematical Alphanumeric Sym.
2150..218F	Number Forms	20000..2A6D6	CJK Unified Ideographs Ext. B
2190..21FF	Arrows	2F800..2FA1F	CJK Compatibility Ideographs Supp.
2200..22FF	Mathematical Operators	E0000..E007F	Tags
2300..23FF	Miscellaneous Technical	F0000..FFFFD	Private Use
2400..243F	Control Pictures	100000..10FFFFD	Private Use
2440..245F	Optical Character Recognition		

유니코드 표준은 국제표준 ISO/IEC 10646:2003과 완전히 일치한다. 유니코드는 ISO(국제표준화기구)와 미국 소프트웨어 업체를 중심으로 한 유니코드 컨소시엄에 의해 제정되었다. 처음에 ISO와 유니코드 컨소시엄은 각각 4바이트 체계와 2바이트 체계의 문자 표준코드를 추진하였으나 이러한 두 가지 표준코드의 추진으로 인한 혼란 방지와 효율적인 표준코드를 제정하기 위해 두 기구는 연합하게 되었다. 따라서 한 글자를 4바이트로 나타내는 USCS-4(4바이트 부호계)로 시작한 ISO는 UCS 코드체계를 제정함에 있어 UCS-4는 나중에 미루고 2바이트 체계의 UCS-2를 먼저 만드는데 합의하였다. 이런 과정을 통해 유니코드가 제정되었다. 유니코드 즉, ISO/IEC 10646 표준의 이름은 Universal Multiple-Octet Coded Character Set인데, 이를 줄여서 UCS라고 한다.

1.2.2 유니코드의 구조

유니코드의 구조는 크게 17개(1개의 기본언어판(BMP)과 16개의 보충언어판)의 언어판으로 구성되어 있다. 각 언어판은 216, 즉 65,536개의 문자를 지정할 수 있으므로 유니코드에서 지정할 수 있는 문자의 수는 17x65,536=1,114,112개이다. 하지만 대행코드영역(Surrogates) 2,048개를 제외해야 하므로 실제로는 1,112,064개의 문자를 지정할 수 있다. Unicode 3.1 표준에서 각 언어판은 다음과 같이 정의되어 있다.

[표 2] UNICODE 3.1의 각 언어판 정의

언어판	UCS4영역	이름	문자수
0 (0x00)	0x00000000-0x0000FFFF	기본언어판(BMP, Basic Multilingual Plane)	49,196
1 (0x01)	0x00010000-0x0001FFFF	보충언어판(SMP, Supplementary Multilingual Plane for scripts and symbols)	1,594
2 (0x02)	0x00020000-0x0002FFFF	보충표의문자판(SIP, Supplementary Ideographic Plane)	43,253
3 ~ 13 (0x03~D)	0x00030000-0x000DFFFF	아직 사용하지 않음	0
14 (0x0D)	0x000E0000-0x000EFFFF	특별보충판(SPP, Supplementary Special-purpose Plane)	97
15 ~ 16 (0x0F~0x10)	0x000F0000-0x0010FFFF	개인사용목적으로 지정 (Private Use)	0
합 계			94,140

2004년 현재 유니코드는 4.01 베타버전까지 나왔고, 지원하는 문자는 십만자에 이르고 있다.

1.2.3 유니코드 용어의 이해

유니코드 관련해서 자주 언급되는 용어를 아래와 같이 간단하게 정리하였다.

○ 기본언어판, BMP

BMP는 Basic Multilingual Plane의 약자이다. 유니코드의 첫 65,536개의 코드를 의미한다.

○ 언어판, Plane

256x256 즉 65,536개씩의 코드 묶음으로 유니코드에서는 현재 17개의 언어판을 사용할 수 있다. 모두 그룹 00에 포함된다.

○ 언어판 그룹, Group

256개씩의 언어판을 묶어 하나의 그룹으로 명명한다. 유니코드의 17개 언어판은 모두 Group 00에 있다. 유니코드는 17개의 언어판에 한정되어 정의된다. 반면 ISO 표준(UCS-4)에서는 모두 128개의 언어판 그룹이 정의될 수 있다.

1 Plane = 65,536 code points

1 Group = 256 planes = 256x65,536 = 16,777,216 code points

UCS-4 = 128 groups = 128x16,777,216 = 2,147,483,648 code points

○ 인코딩, Encoding

문자세트를 표현하는 방식을 말한다. 유니코드는 코드체계 또는 문자세트를 명명하는 것이며 이를 표현하기 위해서는 UTF-8, UTF-16, UTF-32 등과 같은 인코딩이 필요하다.

○ UCS-2: Universal Character Set 2(octets)

좀더 정확하게는 Universal Multiple-Octet Coded Character Set 2이다. ISO/IEC 10646의 용어로 BMP의 65,536개 코드를 정의하며, 2바이트로 표현

된다. 1개의 언어판, 즉 BMP만이 이에 해당한다. UCS-2는 인코딩 방법이 아니며 문자코드 자체지만, 통상적으로 인코딩으로 사용된다. 여기서 octet이라는 용어를 사용했는데 이 용어는 ISO쪽에서 사용하는 용어로, 유니코드 진영에서 사용하는 바이트와 같은 뜻이다

○ UCS-4: Universal Character Set 4(octets)

ISO/IEC 10646의 용어로 4바이트로 표현된다. 모두 128개의 언어판 그룹, 즉 128×256 언어판 = 32,768개 언어판을 정의한다. 이는 대략 $2^{31} = 2,147,483,648$ 개의 코드에 해당한다. UCS-4는 UCS-2와 마찬가지로 인코딩 방법이 아니며 문자코드 자체이다.

○ UTF-8: UCS Transformation Format, 8-bit form

유니코드 표준의 인코딩 방식 중 하나이다. 표준에서는 17개 언어판의 문자만을 표현할 수 있으나 기술적으로는 UCS-4 전영역의 문자를 표현할 수 있다. 문자에 따라 1~4(또는 6) 바이트로 표현된다.

○ UTF-16: UCS Transformation Format, 16-bit form

각 문자를 16비트로 표현한다. 16비트로 표현한다는 점에서는 UCS-2와 흡사하지만 대행문자영역(Surrogates)을 이용하여 16개의 보충 언어판 코드를 표현할 수 있는 인코딩이다. UCS-2에서는 65,536개의 코드만을 정의할 수 있으나 UTF-16에서는 1백만여자를 더 표현할 수 있다.

○ UTF-32: UCS Transformation Format, 32-bit form

32비트 즉 4바이트로 각 문자를 표현한다. 이점에서 UCS-4와 동일하지만 17개의 언어판만을 정의한다는 점에서는 UCS-4의 부분집합으로 간주하면 된다. UCS-4와 동일하나 0x00000000 ~ 0x0010FFFF 범위만을 합법적인 코드 값으로 갖는다.

2. 국내외 유니코드 체계 적용 사례

2.1 국외

2.1.1 일본 NII

1) 개요

일본 국립정보학연구소(NII, National Institute of Informatics)는 정보학 관련 종합 연구 및 학술정보 유통을 위한 기반 기술 개발 및 정비를 목적으로 2000년 4월에 설립된 기관이다.¹⁾ NII는 목록정보 서비스, 정보검색 서비스, 전자도서관 서비스, 학술정보 네트워크 등의 학술정보 서비스를 제공하고 있다.

2) NII 목록시스템

NII의 목록시스템(NACSIS-CAT)은 1984년 12월에 운용을 시작하였다. 그 후, 하드웨어의 소형화와 고속화를 통한 다운사이징(down sizing)²⁾ 바람이 일어났으며, 소프트웨어 측면에서는 오픈시스템, C/S 시스템(client/server system), GUI(graphic user interface) 조류가 일반적인 것이 되었다. 이에 따라 구 NACSIS-CAT과 ILL 시스템도 변화에 적응하는데 한계가 있다고 판단하였다. 이에 1995년 신(新) CAT-ILL시스템 검토회의를 설치하고, 신규 시스템이 갖추어야 할 주요한 내용에 대하여 논하였다. 그 내용은 다음과 같다.

○ [신(新) CAT-ILL시스템 검토회의] 주요 논의 내용

첫째, 다운사이징과 인터넷 시대에 부응하는 새롭고 유연한 시스템이 필요하다.

1) 1986년 4월에 설립된 문부성 산하의 학술정보센터(NACSIS, National Center for Science Information Systems)를 폐지·전환하여 설립된 기관이다. <http://www.nii.ac.jp/index-j.html>

2) 기업 등에서 사용하는 컴퓨터 시스템을 소형의 것으로 대체하는 것이다. 개인용 컴퓨터(PC)나 워크스테이션 등 소형 컴퓨터의 성능이 현저하게 향상되어 메인 프레임보다 가격 대 성능비가 우수하고, 구내 정보 통신망(LAN) 환경이나 분산 데이터베이스, 트랜잭션 처리 시스템 등 분산 처리에 적합한 환경이 정비되어 있어서 다운사이징을 가능하게 하고 필요하게 하였다. 이전에는 메인 프레임으로 하던 처리를 LAN상에서 PC나 워크스테이션으로 처리하고 있다. 초기 도입 비용의 절감, 개량·확장의 유연성 등의 이점이 있으나 운영과 관리가 어렵다(empas IT 용어사전).

둘째, 급격한 이용자 증가에 따른 고가용성, 고안정성, 고성능 시스템이 필요하다.
셋째, 다국어 문자에의 대응이 필요하다.

○ NII 목록시스템의 다국어 대응

1997년에 [신(新) CAT-ILL시스템 검토회의]에서 제시된 요구사항은 신CAT 시스템(C/S시스템)을 개발하였으나 다국어 대응은 구현되지 않았었다. 왜냐 하면, 그 당시에는 UCS 처리 환경이 정비되어 있지 못하였기 때문이다. 따라서, 종합목록 데이터베이스에는 중국어, 한국어 등의 자료에 대한 등록은 전체적으로 진행되지 못하였다. 전국의 대학도서관에는 약 700만권의 중국어 자료, 약 70만권의 한국어 자료를 소장하고 있었다. 그러나 목록시스템이 다국어를 지원하지 못하였기 때문에, 정보의 유통과 교육, 연구에 큰 지장을 초래하였다.

2000년 1월 전산기 교체와 맞추어서, 데이터베이스에 수록되어 있는 모든 데이터를 UCS로 변환하였다. 그리고 신CAT시스템이 다국어(중국어, 한국어 등)를 지원할 수 있도록 함으로써 각 도서관 소장 자료의 목록 작성이 가능해졌다.

○ 신CAT시스템

다국어 대응의 신CAT시스템은 다국어 데이터를 보존하는 데이터베이스의 변경과 다국어 데이터를 입출력하는 서버의 변경으로 구현된다.

- 목록 소재정보 데이터베이스(종합목록 데이터베이스, 참조파일)의 문자셋을 일본어 EUC에서 UCS로 변경한다.
- 신CAT시스템의 서버에서 클라이언트로 데이터를 보낼 때 문자셋으로 JIS와 GB, UCS도 추가한다.
- 도서관 시스템 측에서는 중국어 전용으로는 GB 대응 클라이언트, 중국어 이외의 문자도 취급하려면 UCS 대응 클라이언트가 필요하다.

○ 데이터베이스의 다국어 대응

데이터베이스 문자셋을 UCS로 변경하는 것으로, 기존의 EXC문자와 UCS 외자(클라이언트 문자코드로 변환할 수 없는 UCS 문자를 말한다. 형식은 ◆

U0x◆ 이다. 이때 0x는 UCS16진 코드를 의미한다)는 UCS로 정의되고 있는 확장 알파벳이나 한자로 변환된다. 중국어 간체자와 번체자, 한글 등 대부분의 문자가 UCS로 정의되어 있으므로 표시가 가능해졌다.

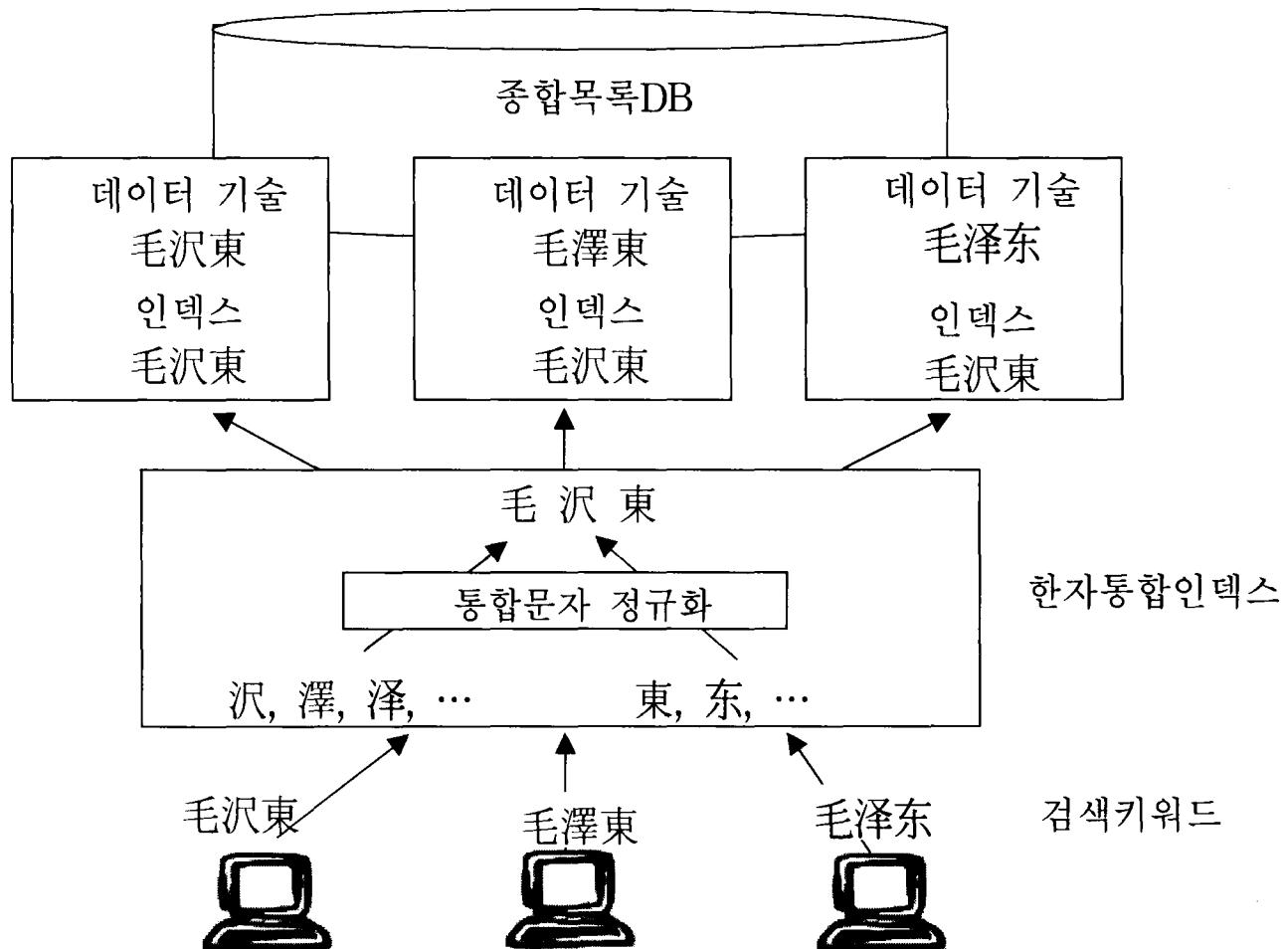
○ 클라이언트의 다국어 대응

UCS에 의한 데이터의 표시와 입력을 위해서는, UCS에 대응하는 CATP 클라이언트가 필요하다. 구체적으로는 윈도우2000 등과 같이 UCS에 대응할 수 있는 OS에서 동작하는 UCS문자세트의 입력을 갖춘 클라이언트를 개발할 필요가 있다. CATP 클라이언트를 개발하고 있는 각 도서관 시스템 업체로 하여금 UCS 대응 클라이언트의 개발을 진행하도록 하고 있다. 또한 NII에서 개발한 UCS 대응 클라이언트 WebUIP를 2001년 1월부터 공개하였다. WebUIP는 웹에 대응한 신CAT/ILL시스템 게이트웨이시스템으로 표준적인 웹브라우저를 CATP 클라이언트로 사용하여 CATP 서버를 이용할 수 있다. 인터넷익스플로러 등의 UCS(UTF8)의 입출력이 가능한 브라우저 환경에서 중국어, 한국어 자료 등의 입력과 디스플레이가 가능해졌다.

○ 한자통합 인덱스

검색기 정규화를 하는 경우, 모양은 다르나 같은 의미의 한자는 그 그룹을 대표하는 한자로 변환한다.

UCS의 한자 통합부분에는 약 2만자의 한자가 포함되어 있고, 모양은 다르지만 같은 의미의 한자가 다수 존재한다. 검색누락을 방지하기 위하여 유사한 형태의 같은 의미의 한자를 포함하여 통합검색이 가능하도록 ‘한자통합인덱스’를 두어 처리하고 있다.



[그림 1] 한자통합인덱스의 구조

2.1.2 미국 OCLC

1) 개요

OCLC에서는 1971년부터 OCLC서비스의 기반이 되고 있는 종합목록 WorldCat의 운용을 시작하였다. WorldCat은 현재 9,000여 회원 기관이 공동으로 목록 작성에 참여하고 있으며, 현재 40여개 언어로 구현된 5,000만건 이상의 서지레코드를 보유하고 있는 세계 최대의 서지데이터베이스이다. OCLC는 종합목록 이용기관들의 다양한 요구를 충족시키기 위해 갱신목록통지(Bibliographic Record Notification)서비스, 자동목록(PromptCat)서비스, 정부문서목록(GovDoc)서비스, 맞춤형편목(Custom Cataloging)서비스, 일괄처리(Batch Processing)서비스 등 각 기관의 특성에 적합한 목록 서비스를 지원하고 있다.

2) OCLC 목록시스템의 다국어 대응

OCLC의 목록 서비스는 WorldCat을 근간으로 하여 제공되는 서비스로 OCLC 목록 서비스의 핵심은 WorldCat을 유지관리하는 종합목록시스템이라고 볼 수 있다. 30여년 전에 운용을 시작한 종합목록 서비스 시스템은 정보 기술 및 환경의 변화에 대응하여 지속적인 성능 개선과 추가 기능 개발을 통해 발전되어 왔으며, 21세기에 걸맞는 차세대의 WorldCat을 구축하기 위하여 전면적인 시스템 전환을 서두르고 있다. 2002년 7월에 OCLC에서 발간한 뉴스레터에 의하면 WorldCat 데이터베이스 플랫폼을 기존의 독자적인 데이터베이스 관리시스템(DBMS)에서 오라클 DBMS로 전환하는 계획을 수립하였으며, 전환된 시스템에서는 MARC 뿐만 아니라 더블린 코어(Dublin Core), TEI, EAD와 같은 여러 메타데이터 형식의 지원은 물론 유니코드를 지원함으로써 다국어에 대응할 수 있도록 하고 있다. 이 계획에 따르면 오라클 플랫폼에서 서비스되는 차세대 WorldCat은 2003년 말에 완성된다.

3) 특징

WorldCat과 Resource Catalog는 ALA 문자세트만이 지원된다. ALA 문자세트는 영어권 국가의 도서관에서 널리 사용되는 문자세트로 로마자 알파벳, 특수문자, 로마자에 기반한 공통적으로 이용되는 발음식별기호를 지원하며, ALA문자세트로 구성된 OCLC MARC 또는 UTF-8 형식의 DC를 반출한다. Connexion 전거파일도 ALA문자세트만을 지원하며 반출은 ALA 문자세트로 구성된 MARC21 형식을 취한다.

Pathfinder는 완전한 유니코드를 지원하며, Dewey 서비스는 유니코드 디스플레이를 지원한다. Connexion은 전세계의 사용자를 위한 웹 기반 시스템으로의 잠재력을 확보하기 위하여 웹 브라우저와의 통신 방식과 내부 저장 형식으로 UTF-8을 사용한다. 아래의 표는 Connexion에서 지원하는 문자세트를 정리한 것이다.

WorldCat은 ALA문자세트로 표현할 수 없는 자료 중 상당부분을 차지하고 있는 한국어, 중국어, 일본어, 페르시아어, 터키어 등의 자료를 입력할 수 있도록 CJK 목록 소프트웨어와 Arabic 목록 소프트웨어를 제공하고 있다. Arabic 목록 소프트웨어는 아라비아어 문자로 작성된 자료에 대한 목록을 지원하는 윈도우 기반의 프로그램이다. CJK 문자를 디스플레이하기 위해서는 윈도우NT, 윈도우2000, 윈도우XP와 같은 유니코드를 지원할 수 있는 OS를 갖추어야 한다.

[표 3] Connexion에서 지원하는 문자 세트

Conexion Db	반입	편집/검증	생성	반출
WorldCat and Resource Catalog	유니코드 변환	ALA문자세트에 대응되는 유니코드	ALA 문자세트	OCLC MARC:ALA 문자세트 DC HTML& DC RDF: UTF-8
Connexion Authority File	N/A	ALA문자세트에 대응되는 유니코드	ALA 문자세트	MARC21: ALA문자세트
Pathfinder DB	유니코드 변환	유니코드	유니코드	UTF-8
WebDewey	N/A	N/A	N/A	N/A

2.2 국내

국내 많은 기관에서 유니코드에 대한 검토와 연구가 이뤄지고 있으나 유니코드 기반의 시스템은 아직은 보편화되지 못하고 있고, 실질적으로 대학도서관에서 도입한 사례도 찾아보기 어렵다.

도서관 시스템이 완전한 유니코드 체계를 지원하기 위해서는 소장 자료의 정보가 유니코드로 저장이 가능하고, 유니코드로 입력, 편집, 검색이 가능하여야 한다. 또한, 목록, 상호대차, 원문서비스 등 각종 도서관 서비스가 유기적으로 연동되어야 하고, 외부 도서관과의 정보교환을 위해서는 유니코드 MARC 반입, 반출을 지원해야 한다. 그러나 아직 유니코드 문자셋을 지원하는 KORMARC가 정식으로 발표되지 않았다. 따라서 국내에 완전한 유니코드 체계를 가진 도서관 시스템이 존재하지 않는 이유는 당연한 것일 수 있다. 현재 유니코드 기반 기술을 도입하여 운영하고 있는 국립중앙도서관, 한국외국어대학교, 한국역사정보통합시스템 등 국내의 유니코드 기반 시스템 구축 사례를 간단히 살펴보겠다.

2.2.1 국립중앙도서관

1) 개요

국립중앙도서관은 2003년 5월 정보시스템에 대한 유니코드 체계로의 전환을 위한 정보화 전략 계획(ISP)에 착수하여 유니코드 체계로의 전환에 따른 문제점과 해결 방안을 모색하고 있다. 그리고 유니코드 문자세트를 수용하는 KORMARC에 대한 초안을 2004년 2월에 발표, 10월에 공청회를 개최하였으며, 의견수렴을 거쳐 확정할 계획이다. 2004년 12월에는 '목록·목차DB'만을 대상으로 유니코드 변환을 수행하였고, 2005년 1월에는 이를 위한 유니코드 시스템 전용 홈페이지³⁾를 가동해서 서비스하고 있다.

2) 분석

국립중앙도서관은 전체 데이터를 대상으로 유니코드로 변환한 것이 아니라, KSC 5601 기반의 데이터와 시스템을 유지하면서, 유니코드 도입에 대한 시범 서비스로 사업을 진행한 것이라, 아직 유니코드 데이터 반입 및 국가간 서지 데이터의 상호 교환은 불가능한 상태이다.

2.2.2 한국외국어대학교

1) 개요

한국외국어대학교 도서관은 학교의 특성상 KSC 5601로 표현할 수 없는 다국어 자료들이 타 대학교에 비해서 많은 편이다. 따라서 이들 자료의 목록 작성 및 검색 서비스를 위하여 유니코드 시스템을 도입하였다.

윈도우2000 이상의 운영체제 환경에서 DBMS로 MS-SQL 서버를 이용하고 있으며, 문자세트로는 UTF-8을 사용하고 있다. 유니코드의 입력은 운영체제에서 지원하는 IME를 이용하여 웹을 통한 검색을 수행하고 그 결과를 볼 수 있다. 비로마자 형태의 베트남어, 아랍어, 이란어, 태국어, 힌디어 등 외국어 자료는 10,000여건 정도 보유하고 있으며, 이들 자료에 대한 서지정보가 신규 유니코드 시스템에 구축되어 웹을 통해 검색이 가능하다.

그러나 독어, 불어와 같은 로마자 형태 자료는 일부 음독기호(ex. @, <, >, *, ~,

3) 국립중앙도서관 유니코드 홈페이지 <http://unicode.nl.go.kr/>

#, ^)를 이용하여 우리 도서관과 같이 대체 문자로 입력하고, KSC 5601에서 지원하지 않는 한자는 한글로 입력하고 MARC 890태그의 \$h를 활용하고 있다. 기존의 KSC 5601에서 지원하지는 않지만 불완전하나마 입력과 검색이 가능한 자료는 기존의 도서관 시스템에서 관리하고, 비로마자 형태의 자료는 별도의 유니코드 시스템으로 관리하고 있다.

2) 분석

기존의 도서관 시스템과 연동되지 않고 독립적으로 존재하는 형태로 구축되어 기존 MARC 데이터베이스와 통합관리가 불가능하다. 또한 MARC 레코드를 이용한 서지 정보 교환도 불가능하다. 기존 도서관 시스템과 독립적으로 운영 및 웹검색서비스⁴⁾를 제공하므로 대학도서관 수준의 유니코드 시스템이라고 말하기 어렵다.

2.2.3 한국역사정보통합시스템

1) 개요

2000년부터 유니코드 사업을 시작하였고, 서지DB는 KSC 5601을 유지하면서 원문데이터만 유니코드를 적용하였다. 유니코드가 한자 지원을 많이 하므로 구축 초기부터 도입하였다.

2) 분석

한글2002를 이용하여 데이터를 구축하고 있으며, 유니코드 폰트나 한적입력기를 자체 개발하였다. 유니코드 기반으로 구축되어 있다고 하나 현재 제정된 유니코드 표준 형태를 따르는 것이 아니므로 유니코드의 표준이 제정된 후 데이터 변환은 필수이다. MS Word 2000, 워디안, 한글2002의 경우 유니코드 체계를 따르는 것이 아니고 사용자 정의 영역을 확장하여 KSC 5601에서 지원하지 않는 한자를 지원하는 것으로 유니코드로의 변환은 필수이다. 그러나 윈도우2000 운영체제에서 MS Word 2002로 DB를 구축하는 경우는 유니코드 형태로 데이터가 구축되므로 추후에 유니코드로 변환하지 않아도 된다. 또한 서지DB도 유니코드화하는 계획을 세우고 있어 서지DB의 데이터 정확도가 높아질 것으로 예상된다.

4) 한국의국어대학교 SolarsWeb4.0 다국어검색 <http://203.232.237.8/contents/search.asp>

3. 유니코드 적용의 필요성

3.1 유니코드의 확장성

우리가 표준으로 사용하고 있는 KSC 5601은 최대 8,836자(94*94)만을 지원할 수 있어 다국어 데이터 표현이나 한자 표현에 있어서 그 한계를 갖는데 반해, 유니코드는 세계 각국의 언어를 통일된 방법으로 표현할 수 있도록 제안된 국제적인 코드 규약으로 백만자까지 표현가능하고 확장가능한 문자세트로 국가간 호환성이 매우 뛰어나다.

[표 4] 유니코드와 KSC 5601 비교

구 분	지원 가능 문자	장단점
유니코드	1,000,000자	통합문자 세트로 호환성 및 확장성 우수
KSC 5601	8,836자	다국어 데이터 표현이 어려움

3.2 국가통합서지관리 기관의 유니코드 체계 검토

국립중앙도서관에서는 2003년 10월에 ‘국립중앙도서관 정보시스템 유니코드 체계 구축 정보화전략 보고서’⁵⁾를 출간하였고, 2004년 시범적으로 목차·목록DB를 유니코드로 변환하는 작업을 수행하였으며, 2005년 1월 유니코드 기반의 홈페이지를 새롭게 가동하는 등 유니코드 기반 시스템으로 확장해 나가고 있다.

또한 한국교육학술정보원(이하 KERIS)도 2003년에 ‘학술 데이터베이스의 유니코드 변환 적용에 관한 연구’를 출간하였고, 2005년 현재 유니코드 시스템 도입을 검토하고 있다. 이와 같이 국가통합서지관리 기관들이 유니코드 기반 시스템을 구축하고 있거나 구축 계획을 세우고 있어, 유니코드 도입은 시기상의 문제일 뿐 도서관 시스템이 가야 할 방향이다.

5) 2004년에 국립중앙도서관의 일부시스템 혹은 특정 시스템에 유니코드 체계를 도입하고, 2005년에 시범운영을 하여 2006년에 전면 구축 등 점진적으로 구축할 계획을 수립하였다.

3.3 다국어 자료에 대한 고품질 DB 구축

우리 도서관은 다른 도서관에 비하여 장서구성이 전문학술 자료와 외국 자료의 비율이 상당히 높고, 양질의 서지데이터를 구축하는데 상당히 많은 노력을 기울이고 있다. 그러나 기본적으로 KSC 5601 문자 코드에서 지원하지 못하는 문자들이 상당히 있어 정확한 데이터 입력 자체가 안되고 있다. 향후 우리나라 대표 대학도서관으로 고품질 서지 DB를 작성하고 이를 관리하기 위해서 유니코드 도입은 반드시 이뤄져야 한다.

도서관에서 보유한 외국어 자료 분포 현황은 아래와 같으며, 타 도서관에 비하여 상당히 외국 자료가 많다.

[표 5] 서울대 도서관의 외국어 자료 분포 현황

(2004년 4월 기준)

언어구분	건수(비율)	언어구분	건수(비율)
영어	445,381(39.48)	스웨덴어	108(0.01)
한국어	353,736(31.36)	덴마크어	97(0.01)
일본어	136,231(12.08)	포르투갈어	79(0.01)
독일어	76,770(6.81)	폴란드어	65(0.01)
중국어	37,708(3.34)	투르크멘어	52(0.00)
프랑스어	28,556(2.53)	아랍어	49(0.00)
스페인어	5,315(0.47)	아제르바이잔어	48(0.00)
러시아어	4,873(0.43)	팔리어	42(0.00)
이탈리아어	2,194(0.19)	체코어	38(0.00)
라틴어	1,194(0.11)	인도네시아어	30(0.00)
터키어	880(0.08)	노르웨이어	26(0.00)
헝가리어	272(0.02)	키르기스어	23(0.00)
네덜란드어	240(0.02)	몽골어	22(0.00)
그리스어(고대)	145(0.01)	핀란드어	21(0.00)
그리스어(현대)	67(0.01)	루마니아어	19(0.00)

현재 우리 도서관은 다양한 외국 자료의 데이터 구축에 있어, KSC 5601에서 지원하지 않는 문자를 위해 자관 임의의 방법으로 입력하고 있다. 첫 번째 방법은 한글이나 다른 문자로 대체한 후 작성하여 MARC 레코드의 890 태그를 사용해서 입력하고 있다. 예를 들어 KSC 5601에서는 한자를 4,888자만을 지원하므로 해당 한자

가 없어서 한글로 입력하는 경우가 다수 발생하고 있다. 한국역사정보 통합시스템 구축에서도 보고된 바에 따르면(이남희 2001), 조선왕조실록 DB 구축은 KSC 5601로 입력시 98.5%가 해결되지만 미해결 한자의 종류(字種)는 50%를 상회(빈도수는 1.5%)한다.

[표 6] KSC 5601 미지원 문자를 MARC의 890 태그에 표기

변형문자 표시	내용	건수	비고
890 \$b	한글 고어나 특수문자를 입력	1,160	
890 \$h	한자가 없어서 한글로 입력	21,413	
890 \$r	로마문자로 변형하여 입력	9	
890 \$x	한자 이외의 외국문자를 한글로 입력	200	
계		22,782	

두 번째 방법은 존재하지 않는 문자를 특수 문자나 기호를 통해 대체 입력하고 있다.

[표 7] KSC 5601 미지원 문자의 대체입력 방법

언어구분	특수문자 입력방법	비고
프랑스어	' (악쌍떼귀) → > ' (악쌍그라브) → < c (세디유) → * " (트레마) → @ ^ (악쌍시르콩플레스) → 무시함	
독일어	" (움라우트) → @ ß (에스체트) → ss	
스페인어	' (악센또) → > ñ (뿔드) → * " (디에르시스) → @	
중국어	" (특별발음 부호) → @ □ (한글음이 없는 경우) → ?? / 로마나이즈	
일본어	ー (장음부호) → ^	

세 번째 방법은 로마나이즈를 하는 것이다. 그러나 로마나이즈 방법은 표기 방식이 달라질 수 있으므로 검색시에 정확성을 보장할 수 없다. 예를 들어 중국서의 경우, 우리 도서관에서는 로마나이즈 방법을 Wade-Giles식으로 작성했으나, 국제적인 추세에 발맞춰 2001년 5월부터 Pinyin식으로 변경하였다. Wade-Giles식으로 구축된 로마나이즈 데이터를 Pinyin식으로 바꿔야 할 수량은 대략적으로 서지데이터는 26,500건으로 추산하고 있고 이외에도 관련 전거데이터도 모두 변환해주어야 한다. 이와 같이 로마나이즈 방법은 표기 방식이 바뀌거나, 또는 이용자가 로마나이즈 표기 방식을 정확히 모를 경우 검색 누락을 피할 수 없다.

3.4 국외 기관과 DB 교류 활성화

도서관에서 일서를 정리할 경우에는 일본 NII의 자료를 많이 참고하고 있다. 국립중앙도서관은 일본 NII와의 협약체결로 NII가 보유한 단행본 서지 DB를 인수하였으나 일서 DB 자체가 많이 활용되지 않으므로 이를 KERIS로 이관했고, KERIS에서 DB를 운용 및 서비스하고 있다. 우리 도서관에서는 목록시스템을 통해 KERIS 참조DB와 연계하여 NII의 DB를 이용할 수 있는데, 문제는 KERIS 자체가 유니코드를 지원하고 있지 않으므로, 데이터 깨짐 현상이 심각하게 발생하여 이를 전혀 활용하고 있지 않다는 점이다. 즉 일본의 국가통합서지 기관과 DB 교류가 원활히 이뤄지지 못하고 있다.

또한 미국 의회도서관(Library of Congress)에서는 유니코드 문자세트를 수용하는 MARC21을 제안하여 향후 유니코드 기반의 데이터 구축에 대비하고 있다. 이외에도 2002년도 MARC21의 서지, 소장, 전거레코드를 유니코드로 구축하기로 계획했었고, 2003년 1월에는 유니코드로 1차 테스트 변환하여 변환된 목록에 대한 검증 수행(천이백만 서지, 오백만 전거, 이천만 소장레코드)하였다.

향후 우리 도서관이 이들 외국의 주요 도서관과 다국어 데이터를 교류할 때 데이터의 누락이나 깨짐 현상 없이 반입·반출하기 위해서는 유니코드 도입이 시급하며, 동시에 이를 통해서 고품질의 DB를 생성하고 관리가 가능해진다.

4. 유니코드 변환시 고려사항

도서관의 데이터를 유니코드로 변환시 다음과 같은 사항을 고려해야 한다.

4.1 동형이음 한자 처리

유니코드에서 한자를 CJK Ideographs Area에 배당하면서도, KSC 5601-1987과의 호환을 위해서 CJK Compatibility Ideographs 영역을 마련하고, 그곳에 동형이음 한자를 배치해 놓았다. 따라서 이와 같은 자료를 변환하기 위해서는 두 가지 방법을 고려해 볼 수 있다.

첫째, 동형이음한자의 다른 음가를 가진 코드를 서로 다른 코드 포인트에 배당하여 모두 코드체계에 수용하는 방법으로 아래의 표와 같다. 이 방식은 한자를 정확하게 입력해야하며, 한자의 한글음을 기계적으로 부여 가능하고, 정렬 가능하다는 장점이 있다. 하지만 다른 음으로 검색했을 경우(예를 들어 요산요수(樂山樂水)를 낙산낙수(樂山樂水))는 검색이 되지 않고, 중국·일본 한자와 호환이 안된다는 단점이 있다. 기존에 KSC 5601에서 상이한 음 모두를 코드 체계에 수용한 이유는 한자의 정렬과 해당 한자를 발음에 따라 한글로 변환함에 있어서 모호성을 제거하기 위함이었다.

[표 8] 동형이음한자 처리 (①상이한 음 모두에 코드 부여)

글자	KSC 5601	Unicode	비고
악할 악(惡)	E4C2	60E1	CJK Ideographs(3400~9FFF)
미워할 오(惡)	E7F7	F9B0	CJK Compatibility(F900~FAFF)

둘째, 서로 다른 음 중 대표적인 음 하나만을 코드체계에 수용하는 방식으로, 유니코드 CJK Ideographs Area에 있는 대표음만을 수용하는 것이다. 이 방식은 음가에 상관없이 하나의 한자로 변환하여 추가 비용을 줄일 수 있고, 중국·일본 한자와도 호환가능하다는 장점이 있지만, 발음에 의한 정렬이 힘들고 동일한 문자를 각 나라마다 다르게 인코딩할 수 없다는 단점이 있다. 일본 NII의 경우는 JIS 코드 자체가 대표음 하나에 코드를 부여한 두 번째 방식이었기 때문에 유니코드로 변환하면서 그대로 두 번째 방식을 수용하였다고 한다.

[표 9] 동형이음한자 처리 (②대표음 하나에 코드 부여)

글자	KSC 5601	Unicode	비고
악할 악(惡)	E4C2	60E1	CJK Ideographs(3400~9FFF)
미워할 오(惡)	E7F7	60E1	CJK Ideographs(3400~9FFF)

위의 두 가지 의견 중에서, 국립중앙도서관은 이론적으로는 동형이음한자를 CJK Ideographs Area로 변환하는 것을 추천하면서도, 유니코드 데이터를 KSC 5601로 변환하여 많이 사용하는 향후 몇 년 동안은 CJK Compatibility Ideographs Area와 함께 변환하는 첫 번째 방법을 추천하고 있다. CJK Compatibility는 우리나라에서 유니코드 컨소시엄에 KSC 5601과의 호환을 위하여 필요하다고 요청하여 할당받은 특수 영역으로, 일본과 중국 등 외국에서는 거의 사용하지 않는 영역이다.

[표 10] 동형이음 한자 처리의 장단점

	① 상이한 음 모두에 코드 부여	② 대표음 하나에 코드 부여
장점	<ul style="list-style-type: none">· 한자를 정확하게 입력하며, 한자의 한글음을 기계적으로 부여 가능· 정렬 가능	<ul style="list-style-type: none">· 유니코드에서 표현되는 27,000여 한자 및 두음법칙을 고려하면, 동형이음 한자는 수천개임· 중국, 일본 한자와 호환 가능
단점	<ul style="list-style-type: none">· 요산요수(樂山樂水)를 낙산낙수(樂山樂水)로 입력하면 검색 안됨· 중국, 일본 한자와 호환 안됨	<ul style="list-style-type: none">· 동일한 문자를 필요에 의해 각 나라가 다르게 인코딩 못함· 정렬 불가능

따라서 우리 도서관도 국립중앙도서관의 분석과 마찬가지로, 기본적으로 KSC 5601과 호환을 위해 마련된 코드를 이용하여 두 가지로 변환하는 것이 바람직하다고 생각된다. 또한 첫 번째 안을 채택했을 경우, 발생하는 검색 문제는 현재 검색 재현율을 위해 사용하고 있는 동자이음정규화 테이블을 사용하거나 또는 CJK Compatibility Ideographs Area에 배당되어 있는 코드 포인트들의 수가 361여개 밖에 되지 않으므로 별도의 매핑 테이블을 고안하는 것이 낫다고 생각된다. 이러한

매핑 테이블을 사용하면, 중국·일본과 데이터 교류시 변환도 용이해진다.

4.2 외부기관과 데이터 교류 방안

도서관 시스템을 유니코드로 전환할 경우, 외부 기관과 데이터 교환시 유니코드 데이터를 주고 받을 때는 문제가 생기지 않으나 현재 대부분의 시스템이 KSC 5601로 인코딩된 데이터를 사용하고 있으므로 자료의 손실이 발생할 수 있다.

데이터의 반입시에는 유니코드가 KSC 5601 인코딩 시스템을 포괄할 수 있으므로 특정 필터 프로그램을 두어 유니코드로 자동변환하면 되므로 문제는 없다. 그러나 유니코드 데이터를 외부에 반출할 때 KSC 5601에 없는 문자를 대체 문자로 변환하여 반출해야 한다.

이에 대한 대처방식으로 현 도서관 시스템에서 사용하고 있는 독자적인 대체 문자 표현 규칙을 사용할 수도 있고, W3C의 HTML 문자 엔터티 표현 규약을 따를 수도 있겠다. W3C에서는 문자 엔터티를 &#와 ; 사이에 십진수(decimal)로 기입하도록 제안하고 있다. 예를 들어 HTML 또는 XML에서 한글 ‘가’를 표현하기 위해서는 단순히 ‘가’라고 입력할 수도 있지만, 다음과 같은 방식으로 표현할 수도 있다.

[표 11] W3C의 HTML 문자 엔터티 표현 규약

문자	16진수(hex)	10진수(dec)	엔터티 표현 (entity representation)
가	AC00	44032	가
惡	60E1	24801	惡
A	0041	65	A

이와 같은 W3C의 HTML 엔터티 표준 규약을 따를 경우, 자료들의 규칙적인 변환도 쉽고, 브라우저에 보여주기도 편하며, 이런 변환을 위해서 개발된 공용 컴포넌트를 그대로 이용할 수 있다는 장점이 있다.

4.3 다국어 입력 방식

다국어를 입력하기 위해서는 Microsoft에서 개발한 IME 시스템을 그대로 이용하

는 방법과, 자체적으로 입력 응용프로그램을 개발하는 방법이 있다. 첫째, IME 시스템을 그대로 이용하면 개발 비용이 들지 않지만, 윈도우2000 이상을 지원해야 하고, 폰트가 없으면 입력 불가능하며 환경 설정을 사용자에게 요구한다는 단점이 있다. 둘째, 입력 프로그램을 자체 제작하면 운영체제에 독립적으로 이용할 수 있지만 주기적인 개발비용이 필요하다는 단점이 있다.

[표 12] 다국어 입력 방법 장단점

구분	① IME	② 응용프로그램에서 지원
장점	<ul style="list-style-type: none"> · 개발 비용의 절감 · 업그레이드 불필요 	<ul style="list-style-type: none"> · 운영체제 독립적 인터페이스 · 시스템 제공 폰트 이용 · RISS에 맞게 UI를 제공하여 편리
단점	<ul style="list-style-type: none"> · 윈도우2000 이상 지원 · 폰트가 없을 때 입력 불가능 · 환경 설정 필요 	<ul style="list-style-type: none"> · 개발 비용의 증가 · 주기적인 유지 보수

두 가지 방법 중 다국어 입력기를 독자적으로 개발하는 것은 여러 모로 비용이 많이 드는 일이다. 하지만 IME를 그대로 사용한다는 것은 사용자에게 심각한 불편을 초래할 수 있다. 하지만 일반적으로 사용자는 자신이 찾고자 하는 특정 언어를 많이 사용하는 경향이 있으므로 MS IME 사용이 바람직 할 것이다.

4.4 유니코드 인코딩 방식

유니코드의 인코딩 방식에 관해 국립중앙도서관에서 다음과 같은 의견을 제시하였다. 첫째, UNIX 운영체제 및 UNIX 어플리케이션 프로그램은 UTF-8을 기반으로 하는 것이 안정성이 뛰어나며, 둘째, Microsoft Windows 기반 시스템인 경우 UTF-16을 기본 인코딩 방식으로 사용하는 것이 좋다. 셋째, 데이터베이스 관리 시스템은 각각의 컬럼에 대해 별도의 인코딩 방식을 설정할 수 있으므로 해당 데이터의 종류에 따라 인코딩 방식을 결정해야 하며, 영문을 주로 사용하는 컬럼은 UTF-8, 동양권 언어를 저장하는 컬럼은 UTF-16으로 인코딩하는 것이 하는 것이 바람직하다고 언급하고 있다.

위와 같은 원칙에 동의하면서도, 운영체제의 저장공간이 얼마나 남아 있는냐를 반드시 고려해야 하고 하드웨어의 업그레이드 시기와 잘 맞추어여야 한다.

4.5 유니코드 미지원 한자 처리

현 시스템에서 유니코드에 포함되지 않는 한자를 처리할 수 있는 방법은 없다. 이를 보완하는 방법으로, 한국역사정보통합시스템 및 국제 한자 표준화 기구에서 지정하는 코드 번호 등을 통일되게 부여하여 관리할 수 있는 방법을 마련해야 한다.

4.6 한글음이 없는 한자 입력 방법

유니코드 입력기와 연관된 문제로, 또 다른 입력 방식(부수 및 획수)을 개발해야 하거나, 유니코드 Ideograph들의 공식적인 로마나이즈 이름으로 입력할 수 있다. 나아가 향후 국립국어연구원의 표준화 방향에 반영되도록 의견을 제시해야 한다.

4.7 옛한글 문제

주리정(2001)은 현 유니코드 시스템에 옛한글의 자모만이 들어있고, 이에 대한 선정 기준도 모호하다는 단점을 지적한 바 있다. 옛한글의 음절을 구성하기 위한 자모만이 들어있고, 이에 대해 기본적인 음절구성 원리만이 들어있을 뿐, 이를 출력하기 위한 방법도 아직 마련되어 있지 않다. 따라서 현재 옛한글을 효과적으로 표현하기 위해서는 유니코드 시스템에 옛한글 영역을 할당받기 위해 체계적인 조사 및 연구가 필요하다.

4.8 검색엔진 교체

우리 도서관 시스템에서 운용되고 있는 BRS Search 검색엔진은 유니코드를 지원하지 못하고 있을 뿐만 아니라 국내 판매사의 제품 판매 중지로 기술지원도 미흡한 실정이다. 이에 유니코드를 지원하는 새로운 검색엔진으로 교체되어야 한다.

검색엔진 데이터는 검색 속도 향상을 위하여, DB에서 추출한 자료를 사용하기 때문에 현재 운용중인 KSC 5601 기반의 검색엔진 데이터를 유니코드 체계로 변화시키는 것이 아니라, 유니코드로 변환된 DB로부터 새로 데이터를 생성하는 것이다.

이는 신규로 모든 데이터를 생성함으로써 보다 최적화된 검색엔진 데이터 구성이 가능하기 때문이다.

아래의 표는 유니코드 지원 버전과 인코딩 방식에 따른 검색엔진의 제품별 특징이다.

[표 13] 검색엔진 제품별 특징

제품명	유니코드 지원 버전	지원 인코딩 방식	KSC 5601에 없는 문자
Verity K2	Unicode 3.0	UTF8	색인 구축 가능
다센 21	Unicode 2.0	UTF8	색인 구축 가능
Konan DOCRUZER	Unicode 3.0 (2005.2월부터)	UTF8	색인 구축 가능
Search Formula-1	Unicode 3.0	UTF8	색인 구축 가능

다국어로 입력된 자료를 검색시스템에서 검색하고자 하는 경우 시스템은 이용자로 하여금 유사한 영어자모를 쉽게 검색할 수 있도록 해주어야 한다. 즉 독일어의 경우 ‘U’가 있고 움라우트가 있는 ‘Ü’가 있을 수 있다. 이 때 실제 목록자료는 ‘Ü’로 되어 있는 경우 이용자는 ‘U’를 입력하여 검색을 실행하더라도 해당 목록이 검색될 수 있도록 처리하는 기능을 지원해야 한다. 이렇게 해야 하는 이유는 움라우트를 글자판에서 직접 입력하지 못하고 문자표 등을 이용해야 하는 경우가 대부분이기 때문이다. 따라서 시스템은 ‘Ü’와 ‘U’ 등의 모든 경우를 처리하는 방향으로 개발하여야 한다.

유니코드에서 지원되는 다국어는 80여개에 이른다. 그러나 현실적으로 우리가 시스템을 개발하는데 있어 모든 언어를 다 고려할 수 없다. 따라서 주요 언어를 목표로 개발을 하는 것이 필요하다. 시스템에서 지원한다는 것은 문자의 입력과는 무관하며 입력된 문자를 가지고 색인어를 만들고 검색어를 처리하는데만 영향을 미친다. 즉, 입력된 글자가 문자나 숫자이면 그대로 색인어를 만들고 기호나 특수문자, 공백이면 삭제한다. 또한 라틴계열 언어 중 ‘èàùç’등과 같은 문자는 ‘eauc’등으로 바뀌서 색인어를 생성한다.

모든 다국어의 문자는 입력작업시 기본적으로 OS에서 지원되는 입력기를 이용하

여 입력할 수 있다. 키보드에서 바로 입력할 수 없는 확장문자나 부호, 기호, 특수 문자 등은 ‘문자표 입력기능’을 지원하여 입력작업을 수행하도록 한다.

5. 서울대 도서관 적용 방안

5.1 현 시스템 검토

도서관의 시스템별 유니코드 지원 유무 현황을 아래와 같이 정리해 보았다.

[표 14] 서울대 도서관 시스템 유니코드 지원 유무 현황

구분	내용	수량	유니코드 지원(O/X)	비고
서버	UNIX(Solaris 2.8)	6대	O	
	NT(Win 2000 Ser)	5대	O	
	Linux	2대	O	인터넷 디스크용 및 게시판 서버
클라이언트 PC (업무용*)	Win XP		O	
	Win 2000	20대 (추정)	△	완벽하게 지원 못함
	Win 95/98/ME 등		X	
DBMS	Oracle 9i		O	
검색엔진	BRS Search 6.3		X	
프로그램 관련	Visual Basic 6.0		O	
웹	Tomcat 3.23		O	
	Apache 1.3.12		O	

5.2 유니코드 개발 기본 사항

서울대학교 도서관 정보시스템에 유니코드 체계 적용은 단순한 코드셋 변경의 문제가 아닌, 앞에서 분석한 내용처럼 여러 고려사항이 존재한다. 이 중 자체 해결 가능한 범위 안에서 시스템을 업그레이드하여야 한다.

기존 SOLARSII 시스템의 웹 프로그램은 자바환경이므로 프로그램에서 인코딩 변경만으로도 유니코드 지원이 가능하나 다양한 컨텐츠 수용에 어려움이 있어 재개발이 필요하다. 또한 클라이언트 관리 프로그램은 VB 6.0으로 개발되어 있어 유니코드를 지원하지 않으므로 단순 업그레이드가 아닌 유니코드를 완벽하게 지원하도

록 개발되어야 한다.

현재 운영 중인 전자도서관시스템(비-유니코드)을 동시 운영하기 위해서는 별도의 DBMS 시스템(유니코드 코드셋 환경)이 필요하며, 유니코드가 지원되는 검색엔진 도입이 필요하다. 또한 향후 유니코드 기반의 시스템에서는 SOLARSII와 전자도서관시스템으로 분리 운영되는 것이 아니라 통합적으로 운영되어 통합검색도 원활하게 이루어져야 한다.

5.3 소요 일정 및 예산

유니코드 기반의 시스템 개발에 소요되는 일정과 예산은 기초 연구과제⁶⁾를 참고로 하여 요약하였다.

5.3.1 소요 일정

시스템 분석 및 설계를 시작으로 하여 어플리케이션 업그레이드, DB변환 및 교열, 시범운영까지 10개월~1년 정도 소요될 것으로 추정된다.

[표 15] 유니코드 기반 시스템 개발 소요 일정

구 분	M+1	M+2	M+3	M+4	M+5	M+6	M+7	M+8	M+9	M+10
시스템 분석 및 설계										
운영체제 변환										
어플리케이션 업그레이드										
검색엔진 도입										
업무용PC 업그레이드										
DB 변환 및 교열										
시범운영 및 가동										

6) 서울대학교 도서관 정보시스템 유니코드 체계 구축에 관한 연구(책임연구원 신호필, 서울대학교 언어학과, 2005. 1.)

5.3.2 소요 예산

유니코드로 시스템이 전환되는데 소요 예산은 업무용 PC OS의 업그레이드, 유니코드 지원 검색엔진 도입, 어플리케이션 업그레이드, 용량 확장에 대비한 스토리지 구매, 데이터변환과 교열 등을 포함해서 아래와 같이 추정해 볼 수 있다.

[표 16] 유니코드 기반 시스템 개발 소요 예산 (단위 : 백만원)

항 목	예상 비용	비고
Windows XP 업그레이드 (업무용)	2	20대x0.1백만원 (OS 업그레이드)
유니코드 검색엔진 도입	100	
어플리케이션 업그레이드	521	
스토리지 구매	100	유니코드 변환시 용량 확장
DB변환 및 교열	133	
합 계	856	

1) PC OS 업그레이드

유니코드 기반의 시스템이 원활히 운영되기 위해서는 업무용 PC는 윈도우 XP로 반드시 업그레이드되어야 한다. 다행히도 대부분의 업무용 PC는 XP를 사용하고 있으며, 분관의 일부 PC와 학과 및 연구소 PC 중에서 XP로 OS 업그레이드 되어야 할 PC는 대략 20여대일 것으로 추정된다.

2) 유니코드 검색엔진 도입

도서관에서 현재 검색엔진으로 사용하고 있는 BRS Search는 유니코드를 지원하지 않으므로, 유니코드기반의 시스템에서 구동할 수 있는 검색엔진을 새로 도입해야 한다. 유니코드를 지원하는 검색엔진으로는 Verity K2, 다센 21, Konan Docruzer, Search Formula-1(Korea WISEnut) 등이 있다. 이들 검색엔진 도입시에는 성능테스트(BMT)를 통해 도서관의 데이터의 특성과 이용자 이용행태를 반영할

수 있는 검색엔진을 도입해야 한다.

3) 어플리케이션 업그레이드

어플리케이션 업그레이드 대상은 SOLARSII의 수서시스템, 연속간행물시스템, 기증·교환시스템, 목록시스템, 검색시스템, 대출/반납시스템, 장서관리시스템, 상호대차시스템, 경영정보시스템, 도서비관리시스템, 권한관리시스템, 통합검색시스템, 이용자관리시스템, 기사색인시스템 및 전자도서관시스템을 대상으로 한다.

4) 스토리지

도서관의 전체 데이터가 유니코드로 변환되면서 데이터 사이즈가 2배 이상으로 커질 것으로 예상되는 바, 스토리지가 추가 구매되어야 한다.

[표 17] 메인(DB) 서버용 스토리지 구성

구분			용량	비 고	
EMC Symmetrix 8530	현재	사용	150GB	사용용량 (150GB)	오라클 DB (50GB)
		가용	150GB		색인 (70GB)
		전체	300GB		기타 (30GB)
	유니코드 변환시	전체	400GB	· 현재 DB 사용용량의 2배 이상 필요 (300GB + 추후 DB 증가 고려) · 스토리지에 400GB 디스크 확장 필요	

5) DB 변환 및 교열

SOLARSII 시스템 내의 MARC 서지데이터를 비롯하여, 이용자DB, 수서DB, 예산DB 등 방대한 양의 데이터가 유니코드로 변환되어야 하고 더불어 전자도서관 내의 XML 메타데이터와 관련 운영 데이터들이 모두 유니코드로 변환되어야 한다. 전체 데이터의 일괄 유니코드로의 변환과 아울러 대체문자는 정확한 유니코드 값으로 대체되어야 한다. 기본적으로 프로그램으로 일괄변환을 하고, 이후 교열작업을 통해서 데이터 변환에 따른 오류를 줄여나가야 한다.

[표 18] DB변환의 대상물량 및 예상 비용

구분	대상 물량(건)	작업내용 (일처리건수)	예상 비용(천원)
전체 DB 일괄 변환 ⁱ⁾ · 전체 DB변환 · 대체입력문자 변환	11,493,473		100,000
대체입력문자 교열 ⁱⁱ⁾	250,642	교열 (400건)	24,409
수작업 변환 ⁱⁱⁱ⁾	22,782	입력 및 교열 (100건)	8,874
합 계	11,766,897		133,283

- i) 전체 DB 일괄 변환은 SOLARSII 시스템내 MARC 서지데이터와 관련 운영데이터, 전자 도서관의 XML 메타데이터와 관련 운영 데이터 등 전체적으로 데이터 변환을 위한 프로그램 개발 적용, 교열, 검수작업까지 포함함. 그러나 정확한 대상물량 산출이 어렵기 때문에 산출이 가능한 서지DB, 전거DB, 기사색인DB, 콘텐츠메타DB만을 대상으로하여 서지DB 1,227,676건/ 전거DB 244,496건/ 기사색인DB 9,615,098/ 원문DB 406,203건(2003년 정보화현황책자 참고)으로 추정함
- ii) 대상물량 250,642건은 MARC서지 데이터 중에 프랑스어(28,556건) 독일어(76,770건) 스페인어(5,315건) 중국어(3,770건:전체량 중 10% 예상) 일본어(136,231건)를 추산해서 산출함. 또한 자료의 특성상 해당 언어별 전문가를 채용하여 교열작업을 수행함(교정자 38,955원*25일 기준)
- iii) KSC 5601에서 지원하는 한자가 없어서 한글이나 다른 문자로 대체 작성하여 MARC의 890 태그에 표기된 자료는 22,782건이고, 이 자료는 실제 자료를 확인해서 수작업으로 데이터 입력해야 함

6. 결론

유니코드는 세계 모든 언어를 포괄할 수 있다는 점에서 주요 도서관들이 이를 도입했거나 도입 검토 중에 있다. 유니코드 기반으로 시스템이 구축이 되면, 기존의 KSC 5601 코드 사용으로 표현하지 못한 다양한 외국 문자에 대해 보다 완벽한 정보를 입력하여 이용자에게 정확한 정보를 제공할 수 있고, 국내외 기관과 원활한 정보 유통 및 활용체계를 확립함으로써 이용자에게 보다 풍부하고 다양한 정보 제공이 가능하게 된다. 또한 국외 주요 학술정보센터(OCLC, LC, NII 등)로부터 유니

코드 자료를 데이터의 손실이나 깨짐 현상 없이 그대로 반입이 가능하다. 이를 위해 도서관에서는 2005년부터 유니코드 기반으로 시스템을 개편할 계획을 갖고 있다. 현재 기초 연구 및 분석 자료로 연구과제가 이루어지고 있으며 이를 토대로 좀더 전문적인 분석과 주요 정책을 결정할 것이다. 또한 국내외 사례를 연구하여 오류를 최소화하고, 유니코드가 갖고 있는 많은 장점을 최대한 수용할 수 있도록 설계를 할 것이며, 기존의 KSC 5601 기반의 시스템을 유지하면서 충분한 시범운영을 한 후 실제 시스템을 가동하여 최적의 환경을 마련할 것이다. 우리 도서관이 대학도서관 선두에 서서 새롭고 창의적인 프로젝트를 수행한 경험을 토대로 대학도서관에서는 처음으로 시도되는 유니코드 기반의 도서관 시스템 구축 사업이 성공적으로 이루어지기를 기대하며, 우리 도서관인들도 유니코드에 대해 좀더 많은 관심을 가지고 참여해야겠다.

◆ 참고문헌 ◆

- Unicode, Inc. 2005. Unicode Homepage. [cited 2005. 1. 6]. <<http://www.unicode.org/>>
- 국립중앙도서관. 2003. 국립중앙도서관 정보시스템 유니코드 체계 구축 정보화 전략 보고서
- 국립중앙도서관. 2005. 유니코드홈페이지. [cited 2005. 2. 7]. <<http://unicode.nl.go.kr>>
- 신효필. 2005. 서울대 도서관 정보시스템 유니코드 체계 구축에 관한 연구
- 이남희. 2001. 『한국 기록 유산 디지털화의 현재와 미래- 조선왕조실록 사례를 중심으로』. International Seminar on Memory of the World
- 이상로. 2004. 코드와 한글. [cited 2005. 2. 7]. <<http://trade.chonbuk.ac.kr/~leesl/code/>>
- 조순영. 2003. 학술 데이터베이스의 유니코드 변환 적용에 관한 연구
- 주리정. 2001. 『유니코드의 구조와 문제점』. 제8회 한국정보관리학회 학술대회 논문집, 23-28.
- 한국외국어대학교. 2005. SolarsWeb 4.0 다국어검색 (UNICODE Search). [cited 2005. 2. 7]. <<http://203.232.237.8/contents/search.asp>>